Assignments on String Class

1.Write an application to determine the length of the String str =”Hello World”.

**public** **class** StringLength{

**public** **static** **void** main(String args[])

{

String s1="Hello world";

System.***out***.println("length of string : "+*getLengthOfStringWithCharArray*(s1));

}

**public** **static** **int** getLengthOfStringWithCharArray(String str)

{

**int** count=0;

**char**[] strCharArray=str.toCharArray();

**for**(**char** c:strCharArray)

{

count++;

}

**return** count;

}

}

Output :

length of string : 11

2.Write a application to join the two strings “Hello”,”How are you ?”

**public** **class** StringConcat {

**public** **static** **void** main(String[] args) {

String s1="Hello!";

String s2=" How are you ?";

String s3 = s1.concat(s2);

System.***out***.println(s3);

}

}

output :

Hello! How are you ?

3. Given a string “Java string poool refers to collection of strings which are stored in heap memory” . Perform following operations.

a.lowercase

b.uppercase

c.replace all ‘a’ character with ‘$’

d.check if original string contains the word “Collection”

e.Check if given string matches with new string

f.Use another method to match 2 strings

**public** **class** StringMethods

{

**public** **static** **void** main(String []args)

{

String s1="Java String pool refers to collection of Strings which are

stored in heap memory";

System.***out***.println(s1);

System.***out***.println("");

String s2=s1.toLowerCase();

System.***out***.println("Lowercase : "+s2);

System.***out***.println("");

String s3= s1.toUpperCase();

System.***out***.println("Uppercase : "+s3);

System.***out***.println("");

String s4=s1.replace('a','$');

System.***out***.println("Replaced : "+s4);

System.***out***.println("");

**if**(s1.contains("collection"))

{

System.***out***.println("collection is available in string");

}

**else**

{

System.***out***.println("not available");

}

System.***out***.println("");

String s5="Java String pool refers to collection of Strings which

are stored in heap memory";

**if**(s1.equals(s5))

{

System.***out***.println("String5 Matches with string1");

}

**else**

{

System.***out***.println("Doesn't matches");

}

System.***out***.println("");

**if**(s1.compareTo(s5)==0)

{

System.***out***.println("String1 compares with string5");

}

**else**

{

System.***out***.println("Comparison can't happen");

}

}

}

Output :

Java String pool refers to collection of Strings which are stored in heap memory

Lowercase : java string pool refers to collection of strings which are stored in heap memory

Uppercase : JAVA STRING POOL REFERS TO COLLECTION OF STRINGS WHICH ARE STORED IN HEAP MEMORY

Replaced : J$v$ String pool refers to collection of Strings which $re stored in he$p memory

collection is available in string

String5 Matches with string1

String1 compares with string5

Assignment on StringBuffer class

1.write an application to append following strings “StringBuffer”, “is a peer class of string”, “that provides much of”, “the functionality of strings” using stringBuffer.

**public** **class** StringBuff {

**public** **static** **void** main(String[] args)

{

StringBuffer s1 = **new** StringBuffer("StringBuffer");

StringBuffer s2 = **new** StringBuffer(" is a peer class of string");

StringBuffer s3 = **new** StringBuffer(" that provides much of");

StringBuffer s4 = **new** StringBuffer(" the functionality of strings");

String s = String.*join*("",s1,s2,s3,s4);

System.***out***.println(s.toString());

}

}

Output :

StringBuffer is a peer class of string that provides much of the functionality of strings

2.Insert the following string “insert text” into a string “It is used to \_\_ at the specific index position” at denoted by sign \_\_.

**public** **class** StringBuffInsert {

**public** **static** **void** main(String[] args)

{

StringBuffer str = **new** StringBuffer("It is used to at the specified index position");

StringBuffer str1 = **new** StringBuffer("insertText");

System.***out***.println("string = " + str1);

// insert boolean value at offset 8

str.insert(14, str1);

// prints stringbuffer after insertion

System.***out***.print("After insertion = ");

System.***out***.println(str.toString());

}

}

Output :

string = insertText

After insertion = It is used to insertText at the specified index position.

3.Reverse the given string using StringBuffer

**import** java.lang.\*;

**public** **class** StringBuffRev {

**public** **static** **void** main(String[] args)

{

StringBuffer s1 = **new** StringBuffer ("This method returns the

reversed object on which it was called");

System.***out***.println("String buffer = " + s1);

s1.reverse();

System.***out***.println("String buffer after reversing = " + s1);

}

}

Output :

String buffer = This method returns the reversed object on which it was called

String buffer after reversing = dellac saw ti hcihw no tcejbo desrever eht snruter dohtem sihT

Assignments on StringBuilder Class

1.write an application to append following strings “StringBuffer”, “is a peer class of string”, “that provides much of”, “the functionality of strings” using stringBuilder.

**public** **class** StringBuild {

**public** **static** **void** main(String[] args)

{

StringBuilder s1 = **new** StringBuilder("StringBuilder");

StringBuilder s2 = **new** StringBuilder(" is a peer class of string");

StringBuilder s3 = **new** StringBuilder(" that provides much of");

StringBuilder s4 = **new** StringBuilder(" the functionality of strings");

String s = String.*join*("",s1,s2,s3,s4);

System.***out***.println(s.toString());

}

}

Output:

StringBuilder is a peer class of string that provides much of the functionality of strings.

2.Insert the following string “insert text” into a string “It is used to \_\_ at the specific index position” at denoted by sign \_\_.

**public** **class** StringBuildInsert {

**public** **static** **void** main(String[] args)

{

StringBuilder str = **new** StringBuilder("My name is Alagi");

StringBuilder str1 = **new** StringBuilder("Abhishek");

System.***out***.println("string = " + str1);

// insert boolean value at offset 8

str.insert(12, str1);

// prints stringbuffer after insertion

System.***out***.print("After insertion = ");

System.***out***.println(str.toString());

}

}

Output:

string = Abhishek

After insertion = My name is Abhishek Alagi

3.Reverse the given string using StringBuilder

**public** **class** StringBuildRev {

**public** **static** **void** main(String[] args)

{

StringBuilder s1 = **new** StringBuilder ("This method returns the reversed object on which it was called");

System.***out***.println("String Builder = " + s1);

// Here it reverses the string buffer

s1.reverse();

System.***out***.println("String builder after reversing = " + s1);

}

}

Output:

String Builder = This method returns the reversed object on which it was called

String builder after reversing = dellac saw ti hcihw no tcejbo desrever eht snruter dohtem sihT